AMENDMENTS TO THE CLAIMS:

This listing of claims replaces all prior versions and listings of claims in the application:

LISTING OF CLAIMS:

1. (Currently Amended) A method, performed by at least one processing device, for ~~of~~ simulating a logic design comprised of combinatorial logic and state logic, the method comprising:

representing the combinatorial logic and the state logic using separate graphic elements; ~~and~~

identifying clock domains for the combinatorial logic and the state logic using the separate graphic elements;

generating computer code that simulates operation of portions of the logic design, the computer code being generated based on the clock domains; and

associating the computer code ~~that simulates portions of the logic design~~ with graphic elements that correspond to the portions of the logic design ~~a graphic element that represents the combinatorial logic and with a graphic element that represents the state logic~~.

2. (Original) The method of claim 1, further comprising:

performing an error check on the graphic elements to determine if a single graphic element represents both combinatorial logic and state logic; and

issuing an error message if the single graphic element represents both combinatorial logic and state logic.

3. (Currently Amended) The method of claim 1, further comprising:

generating intermediate code that simulates the portions of the logic design; ~~and~~ ~~generating~~ wherein the computer code is generated from the intermediate code.

4. (Currently Amended) The method of claim 1, wherein the computer code comprises ~~one of~~ C++ ~~and~~ or Verilog.

5. (Currently Amended) The method of claim 4, wherein, if the computer code comprises C++, the method further comprises running the computer code through a cycle-based simulator to provide a simulation of ~~the~~ an operation of the logic design.

6. (Canceled)

7. (Currently Amended) The method of claim 1, further comprising:

generating a topology of the logic design based on the graphic elements;

~~obtaining~~ wherein the clock domains are identified from the topology~~; and~~

~~generating the computer code based on the clock domains~~.

8. (Currently Amended) The method of claim 1 ~~7~~, further comprising:

dividing the computer code into segments ~~based on the logic cones~~; and

compiling the segments separately.

9. (Original) The method of claim 1, wherein state elements comprise elements which hold a particular logic state for a period of time and combinatorial logic elements comprise elements which combine two or more states to produce an output.

10. (Original) The method of claim 1, wherein the graphic elements comprise block diagrams.

11. (Currently Amended) An article comprising ~~a~~ one or more machine-readable ~~medium~~ media which ~~stores~~ store executable instructions to simulate a logic design comprised of combinatorial logic and state logic, the instructions causing a machine to:

represent the combinatorial logic and the state logic using separate graphic elements; ~~and~~

identify clock domains for the combinatorial logic and the state logic using the separate graphic elements;

generate computer code that simulates operation of portions of the logic design, the computer code being generated based on the clock domains; and

associate the computer code ~~that simulates portions of the logic design~~ with graphic elements that correspond to the portions of the logic design ~~a graphic element that represents the combinatorial logic and with a graphic element that represents the state~~

~~logic~~.

12. (Original) The article of claim 11, further comprising instructions that cause the machine to:

perform an error check on the graphic elements to determine if a single graphic element represents both combinatorial logic and state logic; and

issue an error message if the single graphic element represents both combinatorial logic and state logic.

13. (Currently Amended) The article of claim 11, further comprising instructions that cause the machine to:

generate intermediate code that simulates the portions of the logic design; ~~and~~ ~~generate~~ wherein the computer code is generated from the intermediate code.

14. (Currently Amended) The article of claim 11, wherein the computer code comprises ~~one of~~ C++ ~~and~~ or Verilog.

15. (Currently Amended) The article of claim 14, wherein, if the computer code comprises C++, the article further comprises instructions that cause the machine to run the computer code through a cycle-based simulator to provide a simulation of ~~the~~ an operation of the logic design.

16. (Canceled)

17. (Currently Amended) The article of claim 11, further comprising instructions that cause the machine to:

generate a topology of the logic design based on the graphic elements;

~~obtain~~ where the clock domains are identified from the topology~~; and~~

~~generate the computer code based on the clock domains~~.

18. (Currently Amended) The article of claim 11 ~~17~~, further comprising instructions that cause the machine to:

divide the computer code into segments ~~based on the logic cones~~; and

compile the segments separately.

19. (Original) The article of claim 11, wherein state elements comprise elements which hold a particular logic state for a period of time and combinatorial logic elements comprise elements which combine two or more states to produce an output.

20. (Original) The article of claim 11, wherein the graphic elements comprise block diagrams.

21. (Currently Amended) An apparatus for simulating a logic design comprised of combinatorial logic and state logic, the apparatus comprising:

[[a]] memory that stores executable instructions; and

a processor that executes the instructions to:

represent the combinatorial logic and the state logic using separate graphic elements; ~~and~~

identifying clock domains for the combinatorial logic and the state logic using the separate graphic elements;

generating computer code that simulates operation of portions of the logic design, the computer code being generated based on the clock domains; and

associate the computer code ~~that simulates portions of the logic design~~ with graphic elements that correspond to the portions of the logic design ~~a graphic element that represents the combinatorial logic and with a graphic element that represents the state logic~~.

22. (Original) The apparatus of claim 21, wherein the processor executes the instructions to:

perform an error check on the graphic elements to determine if a single graphic element represents both combinatorial logic and state logic; and

issue an error message if the single graphic element represents both combinatorial logic and state logic.

23. (Currently Amended) The apparatus of claim 21, wherein the processor executes the instructions to:

generate intermediate code that simulates the portions of the logic design; ~~and~~ ~~generate~~ wherein the computer code is generated from the intermediate code.

24. (Currently Amended) The apparatus of claim 21, wherein the computer code comprises ~~one of~~ C++ ~~and~~ or Verilog.

25. (Currently Amended) The apparatus of claim 24, wherein, if the computer code comprises C++, the apparatus further comprises executes instructions that cause the machine to run the computer code through a cycle-based simulator to provide a simulation of ~~the~~ an operation of the logic design.

26. (Canceled)

27. (Currently Amended) The apparatus of claim 21, wherein the processor executes the instructions to:

generate a topology of the logic design based on the graphic elements;

~~obtain~~ wherein the clock domains are identified from the topology~~; and~~

~~generate the computer code based on the clock domains~~.

28. (Currently Amended) The apparatus of claim 21 ~~27~~, wherein the processor executes the instructions to:

divide the computer code into segments ~~based on the logic cones~~; and

compile the segments separately.

29. (Original) The apparatus of claim 21, wherein state elements comprise elements which hold a particular logic state for a period of time and combinatorial logic elements comprise elements which combine two or more states to produce an output.

30. (Original) The apparatus of claim 21, wherein the graphic elements comprise block diagrams.